ZeroPrompt: Scaling Prompt-Based Pretraining to 1,000 Tasks Improves Zero-Shot Generalization

Hanwei Xu*, **Yujun Chen***, **Yulun Du***,
Nan Shao, Yanggang Wang, Haiyu Li, Zhilin Yang†
Recurrent AI
{xuhanwei, chenyujun, duyulun, kimi_yang}@rcrai.com

Abstract

We propose a multitask pretraining approach ZeroPrompt for zero-shot generalization, focusing on task scaling and zero-shot prompting. While previous models are trained on only a few dozen tasks, we scale to 1,000 tasks for the first time using real-world data. This leads to a crucial discovery that task scaling can be an efficient alternative to model scaling; i.e., the model size has little impact on performance with an extremely large number of tasks. Our results show that task scaling can substantially improve training efficiency by 30 times in FLOPs. Moreover, we present a prompting method that incorporates a genetic algorithm to automatically search for the best prompt for unseen tasks, along with a few other improvements. Empirically, ZeroPrompt substantially improves both the efficiency and the performance of zero-shot learning across a variety of academic and production datasets.

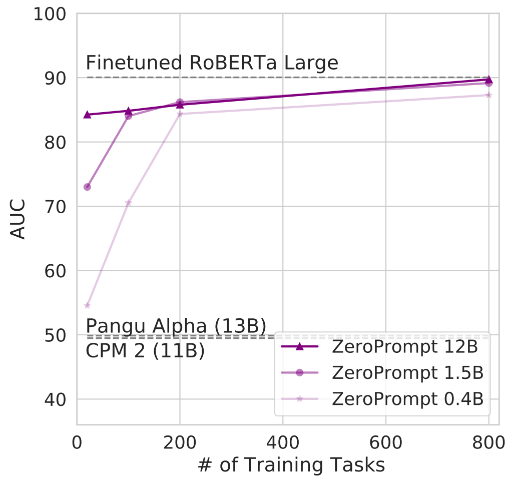

Figure 1: Task scaling vs. model scaling. With an extremely large number of training tasks, the model size has little impact on performance. Moreover, task scaling consistently improves performance at various model scales. For the reference baselines, RoBERTa-Large was finetuned in a fully-supervised manner, while Pangu Alpha and CPM-2 were zero-shot prompted. All models were trained and evaluated in Chinese.

1 Introduction

Pretrained language models, such as BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), T5 (Raffel et al., 2020), and GPT (Radford et al., 2018), are often finetuned for downstream natural language processing tasks, which has been shown to improve performance over non-pretrained models. However, this pretraining-finetuning paradigm still relies on a relatively large set of labeled data for each downstream task to obtain competetive performance. Recent progress like GPT-3 (Brown et al., 2020) demonstrates the possibility of zero-shot and few-shot learning by using prompting on larger-scale models. However, the performance of zero-shot and few-shot generalization still falls short on many tasks compared to fully-supervised finetuning.

To further enhance the performance of zero-shot generalization for pretrained language models, recent work proposed to include a set of supervised tasks into the pretraining procedure and measure performance on unseen tasks during test time (Zhong et al., 2021; Wei et al., 2021; Sanh et al., 2021). This can be viewed as a semi-supervised learning setting where a large unlabeled corpus and a relatively small amount of supervised data are combined for training a better model. Prompts are often used in the framework to unify the tasks. Specifically, Zhong et al. (2021) converted different datasets into a unified "yes"/"no" question answering format with annotated label descriptions. Wei et al. (2021) further extended the scope by considering more task types and a larger model. Sanh et al.

∗ Equal contribution
† Corresponding author

(2021) collected a large set of diverse prompts for each task to further enhance performance.

Despite the recent efforts, there remain a few critical challenges for zero-shot learning. First off, while the effects of scaling some of the dimensions have been explored, such as the model size (Wei et al., 2021) and the number of prompts (Sanh et al., 2021), it is not clear how scaling the number of training tasks affects the performance and the efficiency of multi-task pretraining. We hypothesize that task scaling plays an important role in training generalizable zero-shot systems and explore the limits of task scaling using 1,000 tasks. Interestingly, our empirical study reveals that task scaling can be an efficient alternative to model scaling, as shown in Figure 1. On the one hand, with an extremely large number of training tasks, the model size has little impact on performance. On the other hand, task scaling is more efficient than model scaling, as we show that a 0.4B model can achieve comparable zero-shot performance to that of a 12B model, improving training efficiency by 30 times in terms of FLOPs.

The second challenge is how to obtain high-performing prompts for new tasks. As pointed out by previous work (Liu et al., 2021; Gao et al., 2021a), using different prompts often results in large variance in performance and manually-written prompts are often suboptimal. This challenge is even more critical in a zero-shot setting because it is difficult to reuse training prompts or perform validation. To address this challenge, we propose a novel Genetic Prompt Search (GPS) algorithm that gradually mutates the prompts with a generative model and selects candidates based on performance on a development set. This evolutionary procedure relies on a tiny set of labeled data, only used for validation but not training. We term this setting "zero-shot adaptation with few-shot validation". Although this setting is different from some of the prior works, we believe it is a more practically reasonable setting because it will be difficult, if not impossible, to deploy a model in real-world applications without using at least a few samples for validation. Our Genetic Prompt Search algorithm enjoys the benefit of automatically finding high-performing prompts with only a few validation samples and improved performance compared to manual prompts.

In addition to task scaling and genetic prompt search, ZeroPrompt also explores several other possibilities such as prompt design for multi-task learning. Empirically, ZeroPrompt obtains substantial improvement over both strong zero-shot learning and fully-supervised baselines. Importantly, on many of the production datasets, ZeroPrompt achieves performance better than or comparable to fully-supervised RoBERTa-Large models, a strong baseline that is served in production, which demonstrates the potential for enabling zero-shot learning capabilities in real-world applications.

Our contributions can be summarized as follows.

- We scale the number of tasks to 1,000 in multi-task pretraining for the first time. Our study reveals a new observation that with an extremely large number of tasks, model size has little impact on performance.
- We propose a multitask prompted pretraining approach ZeroPrompt that employs an automatic prompt search for new tasks.
- Our thorough experiments demonstrate that task scaling and genetic prompt search improve both the efficiency and the performance of zero-shot learning.

2 Related Work

2.1 Prompting

Recently, prompt-based learning has been widely explored as it can perform few-shot or even zero-shot learning on many tasks (Schick and Schütze, 2021; Gao et al., 2021a; Le Scao and Rush, 2021). Prompt-based learning bridges the gap between pretraining and finetuning objectives by stitching the text input X with a prompt template and augmenting the label output y as a text string, such that the input and the output can be constructed in a sentence completion task form. Previous methods can be categorized into two types, discrete prompting (Shin et al., 2020; Gao et al., 2021a) and continuous prompting (Liu et al., 2021; Han et al., 2021): The former requires users to provide handcrafted templates (Shin et al., 2020; Le Scao and Rush, 2021), while the latter inserts learnable parameters in the template and directly optimize them during training (Han et al., 2021; Liu et al., 2021; Lester et al., 2021).

In the zero-shot learning setting, discrete prompting requires human efforts to provide manual prompts. Additionally, the settings from previous work where no validation set is used to evaluate task performance are not practical to production NLP systems which require robustness. Continuous prompting requires updating the learnable parameters using labeled data for the tuning phase and thus suffers from requiring even more human efforts in labeling. Instead, our prompting method targets better zero-shot generalization automatically with a few labeled samples per task only used for validation.

2.2 Pretraining and Zero-shot Generalization

Pretrained language models, like BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), T5 (Raffel et al., 2020) and GPT (Brown et al., 2020; Radford et al., 2018; Zhang et al., 2021), have achieved strong performance on various NLP tasks. In some cases, pretrained models can perform well with only a few training samples (Liu et al., 2021; Schick and Schütze, 2021), or even without any training sample (Shen et al., 2021; Sanh et al., 2021). These works mainly focus on unsupervised pretraining on a large-scale corpus.

It has been shown that augmenting unsupervised pretraining with supervised data can significantly improve task performance during finetuning in fields like computer vision and NLP (Chen et al., 2020; Gururangan et al., 2020). Some recent studies followed this idea and obtained improved few-shot or zero-shot generalization in the same manner during pretraining. CROSSFIT (Ye et al., 2021) introduced a repository of diverse few-shot text-to-text tasks to study cross-task generalization. FLAN (Wei et al., 2021) applied instruction-tuning of many tasks to a large-scale decoder model with 137B parameters, while T0 (Sanh et al., 2021) trained an encoder-decoder model with 11B parameters on a large number of professionally crowd-sourced prompts. Our proposed ZeroPrompt also utilizes labeled data in the pretraining phase, but we aim at studying the task scaling law of zero-shot generalization by adopting 1,000 real-world tasks.

2.3 Domain Generalization

Domain generalization addresses the model's ability to generalize to unseen test domains with data from several different but related domains (Wang et al., 2021). Previous approaches in domain generalization are mainly based on gradient operations (Gulrajani and Lopez-Paz, 2020; Arjovsky et al., 2019; Mansilla et al., 2021; Kim et al., 2021) or disentanglement learning (Wang et al., 2020; Peng et al., 2020, 2019). Different from the above

	zero-shot	few-shot	our setting
training	✗	✓	✗
validation	✗	✓	✓
param update	✗	✓	✗

Table 1: Comparison of our "zero-shot adaptation with few-shot validation" setting with previous settings. Zero-shot learning does not rely on any labeled examples either for training or validation, and few-shot learning mostly requires labeled samples for both training and validation, while our setting uses labeled examples only for validation. Only few-shot learning updates the model parameters.

problem formulation, we focus on zero-shot task generalization to unseen tasks with the benefit of prompting to unify different NLP tasks by having the same data format.

3 ZeroPrompt

We follow the same framework of multi-task zero-shot learning in (Wei et al., 2021; Sanh et al., 2021), where models are pretrained on a variety of tasks and then tested zero-shot performance on held-out unseen tasks. We further formulate our problem in the "zero-shot adaptation with few-shot validation" setting, which uses a tiny validation set and is more practical for real-world applications. Under this setting, our ZeroPrompt focuses on task scaling and automatic prompt search.

3.1 Zero-shot Adaptation with Few-shot Validation

We highlight that our "zero-shot adaptation with few shot validation" setting has the practical advantage over others by using a few labeled examples for validation. To disambiguate between different settings, we formally compare our setting with prior zero-shot and few-shot settings, summarized in Table 1.

Many works under the zero-shot setting (Wei et al., 2021; Sanh et al., 2021) simply report the average or the best score of different prompts or models without any procedure for validation, which is not reliable for practical deployment due to skipping the verification step. Another line of works like prompt tuning (Liu et al., 2021) under the few-shot setting rely on labeled examples for both training and validation. Compared to our setting, the computation and deployment costs of few-shot learning systems could be much higher as the models require updating parameters with downstream

Task type	# of Tasks
Sentiment Analysis (**SENTI**)	17 (4,13)
News Classification (**NEWS**)	9 (4,5)
Intent Classification (**INTENT**)	4 (1,3)
Natural Language Inference. (**NLI**)	2 (1,1)
Sentence Similarity. (**STS**)	13 (3,10)
Paraphrase (**PARA**)	1 (0,1)
Question Answer Matching. (**QAM**)	1 (0,1)
Machine Reading Comprehension (**MRC**)	10 (5,5)
Name Entity Recognition (**NER**)	9 (3,6)
Summarization (**SUMM**)	9 (3,6)
Keywords (**KEYS**)	3 (0,3)
Winograd Schema Challenge (**WSC**)	1 (0,1)
App Classification (**APP**)	1 (0,1)
Production tasks (**Objection**)	110 (85,25)
Production tasks (**Profile**)	345 (268,77)
Production tasks (**Execution**)	310 (240,70)
Production tasks (**Mention**)	125 (97,28)
Production tasks (**Violation**)	90 (70,20)
Production tasks (**Acception**)	50 (38,12)
In total	1110 (824,286)

Table 2: The number of tasks for each task type. Numbers in brackets stand for the number of tasks used for training and testing, respectively. e.g. Sentiment Analysis has 4 tasks for training and 13 for testing.

task data.

3.2 Datasets for scaling to 1,000+ tasks

We collected 80 public Chinese NLP tasks from the Chinese Language Understanding Evaluation benchmark (CLUE), the Chinese Natural Language Processing Conference Competition (NLPCC), and many other well-known Chinese NLP competitions. Considering that the number of open-source datasets is still insufficient for studying task scaling, we further acquired over 1,000 real-world datasets from our production systems to investigate the scaling law with the number of tasks. The number of tasks in each task type is listed in Table 2, where we define task types following previous work and intuitive knowledge. The task taxonomy of the production datasets is presented in Figure 2, consisting of 6 task types from 10 different domains. To simulate real-world NLP production systems at scale, where the costs for labeling every data point in a task are expensive, we sample 128 examples per class for each classification task and 256 examples for each generation task[3].

We split the 80 public datasets into 24 training tasks and 56 testing tasks. For the 1,000 production datasets, we randomly reserve 230 tasks for testing and vary the size of the training tasks from 20 to 800 selected in the remaining tasks. Different from FLAN (Sanh et al., 2021) or T0 (Wei et al., 2021), our public test set contains all kinds of task clusters, and the number of test datasets is twice that of train sets for proper evaluation of public tasks. We argue that this is the practical real-world setting with a large number of different unseen tasks in the test set. Detailed train/test split can be found in Table 8.

[3]Only 512 data points are sampled for the iflytek dataset as it has over 100 classes

Figure 2: The task taxonomy of the real-world production datasets. The tasks are collected from commercial sales conversations in ten domains, e.g. *Auto* and *Insurance*. Task types are marked by different colors. For example, "Profile" is a task type to predict an aspect of customer profile from a given transcribed text, and "Acception" is a task type to judge whether a salesperson follows a certain sales script.

Algorithm 1 Genetic Prompt Search

Input: G^0; D_{dev}; f_{GPS}; g_{GPS}; T; K;
Output: Final optimized prompts, G^{T+1}

for each $t \in [0, T]$ **do**
    calculate score for each prompt in G^t using f_{GPS},
    from G^t, select top K prompts as reproductive group G^t_*,
    generate G^{t+1} based on G^t_* using g_{GPS},
end for
from $\{G^0, ..., G^T\}$,select top K prompts as optimal prompts group G^{T+1},
return G^{T+1};

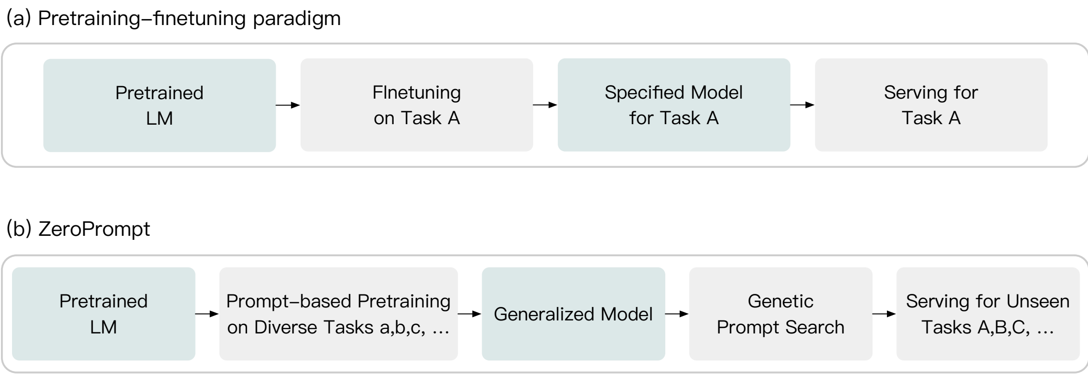

Figure 3: Different NLP paradigms. (a) Traditional pretraining-finetuning paradigm, requiring considerable labeled data for finetuning (b) ZeroPrompt pipeline, only need a tiny validation set for prompt search.

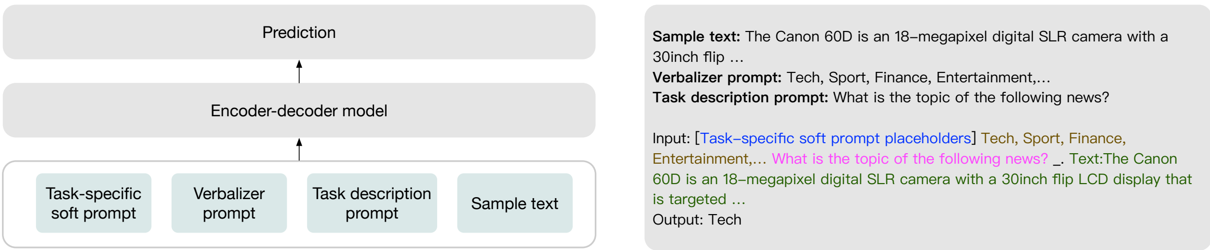

Figure 4: The hybrid prompt composed of task-specific soft prompt, verbalizer prompt and task description prompt.

3.3 Genetic Prompt Search

Previous works on automatic prompt generation are well-studied in the few-shot learning settings (Shin et al., 2020; Gao et al., 2021a), whereas in our paper we focus on the "zero-shot adaptation with few-shot validation" setting. For a new task, we want to automatically obtain high-performing prompts without updating model parameters. Inspired by Genetic Algorithms (Mitchell, 1998), we propose Genetic Prompt Search (GPS) to further improve the zero-shot performance of ZeroPrompt.

Following a more realistic zero-shot learning setting (Keung et al., 2020), we sample a tiny number of data as development set D_{dev} for a downstream task. We term this setting as "zero-shot adaptation with few-shot validation". The process of Genetic Prompt Search is described in Algorithm 1, where f_{GPS} is the metric function to decide which prompt will be reserved or eliminated, and g_{GPS} represents the genetic function to generate new prompts. For a downstream task, the algorithm is firstly initialized with a set of handcrafted prompts G^0. In practice, we can achieve good performance with 3 prompts for initialization. For each iteration, we calculate the scores of prompts in G^t using f_{GPS}, and select the top-K prompts as G^t_*. Then we generate G^{t+1} using g_{GPS} based on G^t_*. Finally, we select the top-K prompts from the whole generated prompt sets.

We experiment with three genetic functions g_{GPS}: **LM-BFF**, according to Gao et al. (2021a), we generate prompts using a pretrained T5; **BT**, we generate prompts using back translation from task specific prompts; **PTMs**, inspired by DINO (Schick and Schütze, 2021), we directly use a large PTM to generate similar prompts.

Figure 3 illustrates the pipeline of ZeroPrompt. Compared to the traditional pretraining-finetuning paradigm, our ZeroPrompt pipeline introduces two key differences. Firstly, a generalized model is served for all downstream tasks, rather than multiple different finetuned models for each specific downstream task. Secondly, we employ Genetic Prompt Search to automatically generate high-performing prompts in our "zero-shot adaptation with few-shot validation" setting. Additionally, our ZeroPrompt uses a hybrid prompt form with both soft prompts and hard prompts.

3.4 Prompt Design

Although large-scale pretrained models with prompting, e.g. GPT-3 (Brown et al., 2020), show promising results on zero-shot generalization to unseen tasks without any labeled data, prompt design is of vital importance to their performance. In this subsection, we describe the prompt design of our choice and some other tested variants.

In the simplest form of a prompt template T, the prompting method constructs T by a handcrafted prompt P and the text input sequence X:

$$T = \{P, X, [\text{MASK}]\} \quad (1)$$

where [MASK] is the blank that an answer should be filled in to complete the sentence. This is known as sentence in-filling.

As illustrated in Figure 4, our optimized prompt P is further decomposed into three parts, $\mathcal{E}$, $\mathcal{V}$, and $\mathcal{D}$, where we have the task-specific soft prompt $\mathcal{E}$, the verbalizer prompt $\mathcal{V}$ and the task description prompt $\mathcal{D}$. As a result, our prompt template T could be expressed as:

$$T = \{\mathcal{E}, \mathcal{V}, \mathcal{D}, X, [\text{MASK}]\} \quad (2)$$

It should be noted that the above prompt design is only a choice from several prompt design variants, whose performance will be reported in 4.3.1. Similar to PET (Schick and Schütze, 2021), we include prompts for generation tasks and prompt-verbalizer pairs (PVPs) for classification tasks. Verbalizer selection is of less importance when the multiple-choice form is applied (see 3.4.1), and thus we choose the commonly-used manual verbalizers for convenience. For each task type, we first design a collection of basic prompts and then modify the basic prompts by adding task-specific keywords or expressions to obtain the final task description prompts. We enforce tasks within each task type to have the same or similar verbalizers for consistency.

3.4.1 Prompts with Verbalizer Candidates

For zero-shot learning on unseen tasks, in which case finetuning is not an option, finding the best-performing verbalizer $v \in \mathcal{V}$ from a large number of vocabulary candidates is a difficult problem. To address this issue, we concatenate all possible verbalizer candidates $\mathcal{V} = \{v_1, v_2, ...\}$ and place the candidates ahead of the task description prompt, as shown in Figure 4. Some prior works (Sanh et al., 2021; Wei et al., 2021) also embraced this multiple-choice prompt form, but no results were reported regarding the zero-shot performance with or without label information. In the ablation studies 4.3.1, we present detailed results and confirm the benefits of utilizing label information in such a prompt form.

3.4.2 Disentangled Task Representations

As mentioned above, we enforce intuitively similar tasks to have similar prompts or PVPs so that task-agnostic knowledge is already properly modeled in this manner. At the same time, task-specific knowledge should be also helpful to zero-shot generalization. To disentangle the task-specific and task-agnostic knowledge in multitask pretraining, we install a continuous prompt embedding as a prefix, which is referred as the task-specific soft prompt shown in Figure 4. For unseen tasks, we need to cold-start and initialize the prompt embeddings from scratch without labeled data. One intuition is to directly use the soft prompts from training tasks with similar data distribution. We experiment with different initialization approaches and the results can be found in Appendix A.3.

4 Experiments

4.1 Experiment Setups

4.1.1 Models

To put the performance of ZeroPrompt in context, we compare our method with several state-of-the-art large-scale Chinese pretrained models with good zero-shot learning performance.

- **Pangu-α** (Zeng et al., 2021): a pretrained decoder model with up to 200 billion parameters. With limited computation resources, we take the 13 billion version of Pangu-α as our baseline.
- **CPM-2** (Zhang et al., 2021): a pretrained encoder-decoder model with 11 billion parameters. We take the Chinese version of CPM-2 as our baseline.
- **RoBERTa** (Liu et al., 2019): We also compare our method with a strong baseline, the fine-tuned RoBERTa-large model.

For our ZeroPrompt, we pretrain an encoder-decoder text-to-text transformer (T5) with 1.5B parameters as our base model. In the unsupervised pretraining stage, the model is pretrained for 100k steps on a 300G web-crawled Chinese corpus with a batch size of 4096 and the sequence length of 512. In the multitask prompted training stage, we train the model with an Adam Optimizer for 1500 more steps with a batch size of 64 and a learning rate of 3.5e-5. We repeat all of our experiments, including multitask prompted pretraining and finetuning, for

task type	task	**CPM-2** Zero-Shot	**Pangu-α** Zero-Shot	**T5** Zero-Shot	**RoBERTa** Finetuning	**ZeroPrompt** Zero-Shot	**T5** Finetuning
SENTI	online_shopping_10cats	80.60	61.99	71.88	$95.30_{(0.42)}$	$95.90_{(0.24)}$	$96.94_{(0.26)}$
	nlpcc2014_task2	68.53	56.22	60.06	$72.09_{(0.80)}$	$80.49_{(0.80)}$	$80.67_{(0.21)}$
	SMP2019_ECISA	29.04	**40.41**	31.21	$69.45_{(1.65)}$	$38.46_{(0.33)}$	$74.15_{(0.30)}$
NEWS	CCFBDCI2020	49.57	38.09	27.48	$90.73_{(0.58)}$	**80.50**$_{(1.68)}$	$96.53_{(0.41)}$
INTENT	catslu_traindev	62.63	46.65	11.27	$91.09_{(2.33)}$	**90.48**$_{(0.78)}$	$94.42_{(0.66)}$
NLI	ocnli_public	33.76	38.58	30.51	$54.70_{(0.53)}$	**46.16**$_{(1.87)}$	$58.15_{(1.61)}$
STS	CBLUE-CHIP-STS	44.15	56.40	44.94	$80.28_{(1.08)}$	**77.90**$_{(0.59)}$	$82.45_{(2.07)}$
	sohu-sts-B-ss	33.50	54.94	43.46	$89.71_{(0.68)}$	**79.85**$_{(1.03)}$	$89.85_{(0.86)}$
QAM	nlpcc2016-dbqa	49.90	56.08	51.69	$56.31_{(1.51)}$	$62.61_{(3.64)}$	$76.76_{(1.95)}$
PARA	PAWS-X	48.08	53.06	48.08	$53.51_{(0.53)}$	$54.90_{(0.37)}$	$59.04_{(0.51)}$
MRC	cmrc2018_public	8.51	11.61	5.94	-	**35.50**$_{(0.73)}$	$61.00_{(0.80)}$
NER	msra_ner	3.11	9.81*	21.44	-	**58.17**$_{(4.40)}$	$65.37_{(2.65)}$
	CMeEE	1.18	9.44*	6.77	-	**24.84**$_{(0.94)}$	$29.34_{(2.84)}$
SUM	EDU_SUMM	1.05	10.02	2.21	-	**14.80**$_{(3.15)}$	$16.97_{(2.11)}$
KEYS	COTE-MFW	1.29	4.91	7.05	-	**50.34**$_{(9.01)}$	$79.35_{(1.08)}$
WSC	cluewsc2020_public	**57.74**	44.93	44.08	$71.99_{(3.32)}$	$47.98_{(4.18)}$	$72.81_{(2.19)}$
APP	iflytek_public	4.77	7.85	1.69	$50.34_{(0.61)}$	**26.14**$_{(1.02)}$	$53.33_{(1.05)}$
Production	Return Commitment	36.28	51.83	53.28	$96.16_{(0.21)}$	**95.53**$_{(0.24)}$	$96.78_{(0.62)}$
	Heating Supply	44.89	31.61	44.57	$97.48_{(0.30)}$	$99.22_{(0.35)}$	$98.91_{(0.59)}$
	Return Amount	53.26	46.09	55.90	$90.71_{(0.33)}$	**89.48**$_{(0.56)}$	$90.86_{(0.47)}$
	Registration Discount	55.09	50.34	56.25	$88.68_{(0.40)}$	**88.48**$_{(0.51)}$	$89.88_{(0.65)}$
	Operation Guidance	57.97	47.71	54.52	$90.78_{(0.35)}$	**78.24**$_{(1.41)}$	$92.80_{(0.84)}$
	Promise for Refunding	46.80	49.35	48.57	$93.71_{(0.24)}$	$94.28_{(0.56)}$	$91.40_{(1.13)}$
	Households Heating Plant	63.37	69.66	48.71	$96.59_{(0.47)}$	$98.22_{(0.52)}$	$97.39_{(0.59)}$
	Refunding Amount	48.48	52.58	49.67	$83.78_{(0.52)}$	$88.03_{(0.83)}$	$83.74_{(1.67)}$
	Cost Abatement	43.18	48.13	51.51	$80.30_{(0.92)}$	$81.88_{(0.22)}$	$81.40_{(1.02)}$
	WeChat Operation	45.45	51.37	47.79	$82.28_{(0.59)}$	**78.25**$_{(0.26)}$	$83.53_{(1.59)}$
AVG		39.71	40.73	37.80	-	**68.76**$_{(1.48)}$	$77.55_{(1.14)}$
AVG excl. GEN		48.05	47.90	44.42	$80.73_{(0.85)}$	**76.04**$_{(1.02)}$	$83.72_{(0.94)}$

Table 3: Main results comparing ZeroPrompt and other zero-shot/finetuning baselines. -: We do not finetune RoBERTa on tagging and generation tasks. *: Only part of the test set is sampled for evaluation due to the computation burden. Red numbers indicate the cases where ZeroPrompt scores better than finetuned RoBERTa and **bold** numbers indicate the cases where ZeroPrompt achieves the best zero-shot performance.

five times with different random seeds to reduce variance.

4.1.2 Metrics

To evaluate the zero-shot generalization performance of our ZeroPrompt and baseline models, we have reserved a diverse set of unseen downstream tasks from both academic and real-world production datasets for testing, including binary classification, multi-label classification, machine reading comprehension, keyword recognition, named entity recognition, and summarization tasks. Here we describe our choice of evaluation metrics for these tasks respectively. Specifically, we use ROC-AUC for binary classification tasks, Micro-F1 for multi-label classification tasks, ROUGE score for summarization tasks, string F1 score for machine reading comprehension and keyword tasks. For NER, we report the positive F1 score, which averages string F1 score between predicted entities and the answer. Details regarding evaluation metrics can be seen in Appendix A.2.

4.2 Main Results

4.2.1 Results of Task Scaling

We explore the limits of zero-shot performance of multitask prompted pretraining, using over 1,000 tasks from 10 domains. Since model scaling has been shown to have significant impact on zero-shot performance (Brown et al., 2020), we also provide the results of three models with 0.4B, 1.5B and 12B parameters. Experimental results with respect

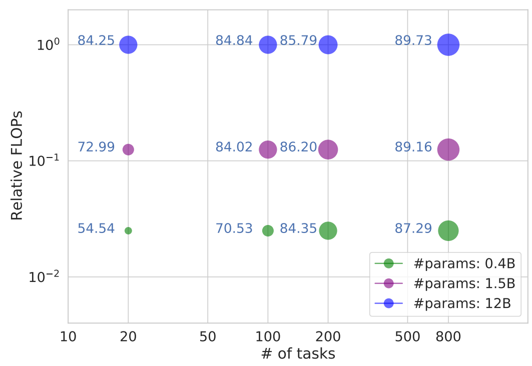

Figure 5: The effect of task scaling on model scaling with respect to zero-shot generalization to unseen tasks. The horizontal axis is the number of training tasks in the multitask prompted pretraining stage; and the vertical axis is FLOPs used in the unsupervised pretraining stage. The model size has marginal impact on performance as more training tasks are added.

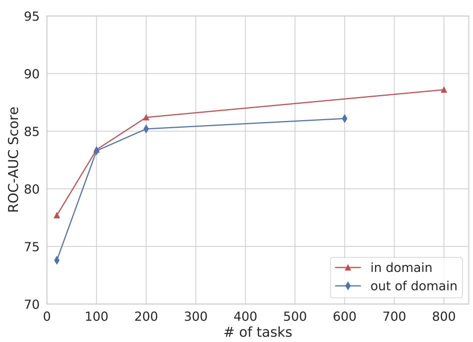

Figure 6: In-domain and out-of-domain zero-shot performance with different number of training tasks.

to task scaling and model scaling are presented in Figure 5. Larger models do have better zero-shot performance with a limited number of training tasks. However, the performance gains from larger models decrease when more training tasks are added. The 0.4B model achieves a score of 87 with 800 training tasks, which scores better than the 12B model with only 200 training tasks. Generally, if we scale the number of training tasks, models can achieve impressive zero-shot performance with small model sizes, substantially improving training efficiency by 30 times in FLOPs (0.4B vs. 12B).

To validate the zero-shot performance on out-of-domain tasks, we select tasks from three domains for testing and the rest for training. Experimental results for the 1.5B model are presented in Figure 6. The zero-shot performance on in-domain tasks rises from 74.7 with 20 training tasks to 88.6 with 800 training tasks, and the performance on out-of-domain tasks rises from 73.7 with 20 training tasks to 86.1 with 600 training tasks. Empirically, task scaling achieves considerable improvement of zero-shot performance on both in-domain and out-of-domain unseen tasks.

	All	Seen	Unseen
baseline	46.16(↑3.89)	46.82(↑2.83)	41.57(↑11.4)
- $\mathcal{V}$	42.88(↑0.61)	43.87(↓0.12)	35.92(↑5.75)
- $\mathcal{E}$	45.06(↑2.79)	46.40(↑2.41)	35.66(↑5.49)
- $\mathcal{E}, \mathcal{V}$	42.27	43.99	30.17

Table 4: Ablation results on the optimized prompt design. baseline: our prompt design with the verbalizer prompt and the task-specific soft prompt; -$\mathcal{V}$: without the verbalizer prompt; - $\mathcal{E}$: without the task-specific soft prompt; - $\mathcal{E}, \mathcal{V}$: without the verbalizer prompt and the task-specific soft prompt.

4.2.2 Results Compared with Other Approaches

In this section, we compare ZeroPrompt with other strong zero-shot and fully-supervised approaches on the unseen testing datasets. Note that due to limited space, only part of the reserved testing tasks, specifically 17 academic and 10 production datasets, are included in Table 3 for comparison.

As shown in Table 3, in the zero-shot setting, ZeroPrompt significantly improves the performance of T5 from 37.80 to 68.76 with a boost of 30.96 points, outperforming previous large PTMs, CPM-2 and Pangu-α, by a large margin of 28 points.

Notably, ZeroPrompt is comparable to or even better than a finetuned RoBERTa-large model on some academic and production datasets. Compared to the overall score of the finetuned RoBERTa, ZeroPrompt is only 4.7 in short. This is quite ecstatic considering that ZeroPrompt did not use any labeled data for tuning. The finetuned T5 is better than RoBERTa, and the gap between ZeroPrompt and the finetuned T5 is less than 8 points.

4.3 Ablation Studies

To understand the importance of optimized prompt design and Genetic Prompt Search in ZeroPrompt, we have performed several ablation studies and reported scores on both the seen task types in the training phase and the unseen task types.

4.3.1 Effect of Prompt Design

We first validate the importance of including the task-specific soft prompt and the verbalizer prompt

	none	weighted avg all samples	weighted avg per sample	top1 per sample	random init
All	44.83	45.76	46.01	46.06	**46.16**
Seen	46.67	46.70	46.77	46.79	**46.82**
Uneen	31.98	39.17	40.65	40.95	**41.57**

Table 5: Ablation results on building new task-specific soft prompt embeddings.

Methods	**Avg**	**Max**	**Min**
Manual Prompt	45.06	45.68	44.44
LM-BFF	43.45 (↓1.61)	44.86 (↓0.82)	42.59 (↓1.85)
LM-BFF†	44.45 (↓0.61)	46.29 (↑0.61)	42.75 (↓1.69)
BT	45.76 (↑0.70)	46.57 (↑0.89)	45.04 (↑0.60)
PTMs (GPT2)	46.03 (↑0.97)	47.25 (↑1.57)	44.86 (↑0.42)
PTMs (CPM)	46.53 (↑1.47)	47.45(↑1.77)	45.56(↑1.12)
BT & GPS (ours)	45.91 (↑0.85)	46.76 (↑1.08)	45.10 (↑0.66)
PTMs (GPT2) & GPS (ours)	46.64 (↑1.58)	47.89 (↑2.21)	45.20 (↑0.76)
PTMs (CPM) & GPS (ours)	**47.05 (↑1.99)**	**48.06 (↑2.38)**	**45.93 (↑1.49)**

Table 6: Ablation results on Genetic Prompt Search. †: we use T5 to generate randomly masked tokens in the manual prompt. GPS: with the Genetic Prompt Search method.

in our choice of prompt design, and then compare different methods to build new task-specific prompt embeddings. Ablation results on the optimized prompt design are shown in Table 4. We can see that task-specific soft prompts and verbalizer prompts are useful when applied separately, and can obtain an even greater gain of 4 points when applied combined by our ZeroPrompt. We also show the ablation results on different approaches to building new task-specific prompt embeddings in Table 5. We can see that the winning approach is surprisingly *random init*, and the direct uses of similar task prompt embeddings seen in training in various ways are slightly worse than *random init*, and the worst performing method is *none* as expected. To comprehend the results on *random init* and *top1*, we suppose that different tasks, though with similar input data distributions, still have different mappings $\mathcal{X}\rightarrow y$. Therefore, it is often difficult to find the most proper task-specific soft prompt seen in the training phase for a new task in the zero-shot learning setting.

4.3.2 Effect of Genetic Prompt Search

In this experiment, we study the effect of Genetic Prompt Search in the “zero-shot adaptation with few-shot validation" setting. To study this problem, we give each task a validation set. For tasks with less than 5 labels or generation tasks, we randomly select 32 samples per task as the validation set; for tasks with more than 5 labels, we randomly select 8 samples for each label.

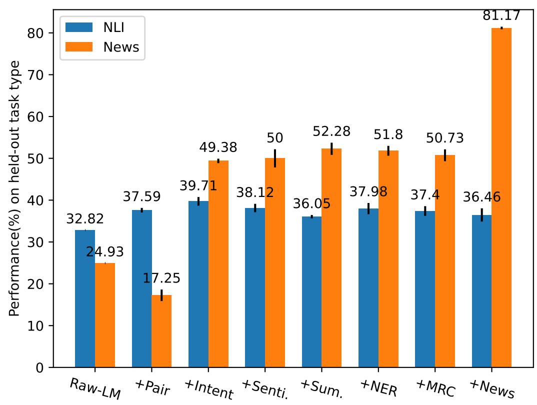

Figure 7: Zero-shot performance on NLI and NEWS with different held-out task types.

In Table 6, 7, we compare three different automatic prompt generation strategies, LM-BFF, BT, PTMs, with and without the proposed Genetic Prompt Search method. We find that the original manual prompts are always suboptimal compared with automatically generated prompts, and the Genetic Prompt Search algorithm contributes to a further improvement. Moreover, in our setting, where only a small validation set is accessible, prompt generation methods with external knowledge, such as BT or PTMs, appear to be superior to those without, like LM-BFF.

4.3.3 Effect of Cross Task Type Transfer

Following the previous works (Wei et al., 2021; Sanh et al., 2021), we study whether held-out task

Methods	weibo_senti_100k	sohu-sts-B-sl	e2e_dials	nlpcc2020-AutoIE	COTE-BD
Manual Prompt	84.23	68.08	82.02	33.95	20.79
LM-BFF	83.17 (↓1.06)	64.75 (↓3.33)	82.77 (↑0.75)	27.12 (↓6.83).	36.80 (↑16.01)
LM-BFF†	83.45 (↓0.78)	66.92 (↓1.16)	83.25 (↑1.23)	36.68 (↑2.72)	38.44 (↑17.65)
BT	84.15 (↓0.08)	68.01 (↓0.07)	82.82 (↑0.80)	36.45 (↑2.50)	38.24 (↑17.45)
PTMs (GPT2)	84.41 (↑0.18)	67.76 (↓0.32)	83.11 (↑1.09)	45.83 (↑11.87)	44.94 (↑24.15)
PTMs (CPM)	84.10 (↓0.13)	68.11 (↑0.03)	84.92 (↑2.91)	44.01 (↑10.05)	44.63 (↑23.83)
BT & GPS (ours)	84.19 (↓0.04)	68.11 (↑0.03)	85.88 (↑3.86)	35.00 (↑1.04)	47.76 (↑26.97)
PTMs (GPT2) & GPS (ours)	83.62 (↓0.61)	67.50 (↓0.58)	83.32 (↑1.30)	49.05 (↑15.09)	49.43 (↑28.64)
PTMs (CPM) & GPS (ours)	**85.03 (↑0.80)**	**68.97 (↑0.89)**	**88.65 (↑6.63)**	**49.14 (↑15.18)**	**49.97 (↑29.18)**

Table 7: Ablation results of several unseen tasks on Genetic Prompt Search. †: only part of the prompt text are generated by T5, the other are still hand-crafted. GPS: with the Genetic Prompt Search method.

types can benefit from multitask prompted pretraining. Specifically, we choose NLI and NEWS as testing task types while other various datasets as training task types. We add different training tasks in sequence as shown in Figure 7. For NEWS, the zero-shot performance increases from 17 to 49 by adding INTENT, while adding sentence pair (STS, QAM, PARA) tasks leads to a performance drop in 7 points. Other training task types such as SENTI, SUMM, NER and MRC only have marginal impacts on the performance. For sanity check, we add NEWS in the training phase at last and the performance increases from 50 to 81 as expected. The zero-shot performance on NLI rises from 32 to 37 by adding more sentence pair tasks, and then to 39 with INTENT, but other training tasks do not further boost the performance. In conclusion, we find that the zero-shot performance on held-out task types can only benefit from some task types, and more labeled data in other task clusters do not always guarantee continuous improvement.

In comparison, our main results on task scaling indicate that performance is boosted when the number of training tasks increases according to the fixed task distribution. Note that task distribution is orthogonal to scaling the task number. How to further improve zero-shot generalization by optimizing task distribution is left to future work.

5 Limitations and Future Work

We show that task scaling, which could now be seen as an alternative to model scaling, improves both the efficiency and performance of zero-shot learning. Our results have a few limitations, however, and it is possible that zero-shot performance could be further improved by studying those problems in the future. Specifically, 1) We mainly focus on in-task type transfer. As shown in ablation 4.3.3, it is still inconclusive if cross-task type transfer can also benefit from task scaling consistently. 2) We control our study by only increasing the number of tasks according to a fixed task distribution. The problem of how to choose a better training task distribution is left for future work, since it is not relevant to our main focus. 3) Our tasks are not exhaustive, because our real-world production data might only represent a subset of all the NLP problems. However, it is challenging to collect more diverse data for the study of task scaling, because publicly available data is often limited in terms of the number of tasks. Therefore, we choose to use production data to initiate such a study. We hope our results could encourage future work on addressing these limitations to further explore the potential of zero-shot learning.

6 Conclusion

In this paper, we propose ZeroPrompt, a multitask prompted pretraining method that significantly improves the zero-shot generalization ability of language models. In our experiments, we collect over 1,000 real-world production tasks to study the task scaling law. We find that the zero-shot performance gap between small and large models becomes less significant when having more training tasks. As a result, task scaling can substantially improve training and serving efficiency. Experiments verified the effectiveness of the proposed Genetic Prompt Search method, which can lead to further performance gain in the "zero-shot adaptation with few-shot validation" setting. We also perform ablation studies on optimized prompt design, verifying the benefits.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ishaan Gulrajani and David Lopez-Paz. 2020. In search of lost domain generalization. In *International Conference on Learning Representations*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification.

Phillip Keung, Yichao Lu, Julian Salazar, and Vikas Bhardwaj. 2020. Don't use english dev: On the zero-shot cross-lingual evaluation of contextual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 549–554.

Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. 2021. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Lucas Mansilla, Rodrigo Echeveste, Diego H Milone, and Enzo Ferrante. 2021. Domain generalization via gradient surgery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6630–6638.

Melanie Mitchell. 1998. *An introduction to genetic algorithms*. MIT press.

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. 2019. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*, pages 5102–5112. PMLR.

Xingchao Peng, Yichen Li, and Kate Saenko. 2020. Domain2vec: Domain embedding for unsupervised domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 756–774. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. *Computing Research Repository*, arXiv:2104.07540.

Feihong Shen, Jun Liu, and Ping Hu. 2021. Conterfactual generative zero-shot semantic segmentation. *ArXiv*, abs/2106.06360.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Guoqing Wang, Hu Han, Shiguang Shan, and Xilin Chen. 2020. Cross-domain face presentation attack detection via multi-domain disentangled representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6678–6687.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. 2021. Generalizing to unseen domains: A survey on domain generalization. In *IJCAI*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. Crossfit: A few-shot learning challenge for cross-task generalization in nlp.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, Chen Li, Ziyan Gong, Yifan Yao, Xinjing Huang, Jun Wang, Jianfeng Yu, Qi Guo, Yue Yu, Yan Zhang, Jin Wang, Hengtao Tao, Dasen Yan, Zexuan Yi, Fang Peng, Fangqing Jiang, Han Zhang, Lingfeng Deng, Yehong Zhang, Zhe Lin, Chao Zhang, Shaojie Zhang, Mingyue Guo, Shanzhi Gu, Gaojun Fan, Yaowei Wang, Xuefeng Jin, Qun Liu, and Yonghong Tian. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation.

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, et al. 2021. Cpm: A large-scale generative chinese pre-trained language model. *AI Open*, 2:93–99.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections.

A Appendix

A.1 Datasets

For fair evaluation of zero-shot generalization, we investigate and collect diverse public Chinese NLP datasets with different task types. The summary of all datasets used in the experiments is presented in Table 8, including train/test task split and metrics of each task. In total, we have 13 task types of public datasets and 6 task types of production datasets.

A.1.1 Public Datasets

- **Sentiment Analysis** requires the model to determine whether the sentiment of a piece of text is positive or negative.

- **News Classification** asks the model to predict the topic of a news article.

- **Intent Classification** asks the model to predict the intent of a person given one of his/her words.

- **Machine Reading Comprehension Question Answering** requires the model to answer a question given a document where the answer can be derived.

- **Natural Language Inference** asks the model to tell the relation of two sentences is neutral, entailment or contradiction.

- **Sentence Similarity** asks the model to predict whether two sentences are similar or not.

- **Paraphrase** asks the model to tell whether two sentences with much lexical overlap are semantically equivalent.

- **Question Answer Matching** asks the model to reason whether the given two sentences can form a valid question answering pair.

- **Name Entity Recognition** requires the model to find all entities in the given piece of text.

- **Summarization** requires the model to give a summary with one or two sentences of the given long document.

- **Keywords** asks the model to extract keywords from the given sentence.

- **Winograd Schema Challenge**, the sample of which composes a sentence, a pronoun and an entity in the sentence, requires the model to tell whether the pronoun refers to the entity.

- **App Classification** asks the model to tell which type of App the given introduction is about, and there are hundreds of target App categories.

A.1.2 Production Datasets

As illustrated in Figure 2, the task taxonomy of our production contains six types of natural language understanding tasks. We provide detailed explanation here and several examples in Table 9.

- **Objection** are datasets that we gathered from production scenario. Objection tasks are language understanding tasks where model will have to analyze whether the speaker is proposing an argument in opposition of the previous contents.

- **Profile** are datasets that we gathered from realistic industrial scenario. Profile tasks are language understanding tasks similar to intent classification, where model will have to tell whether the current sentence is describing certain intention.

- **Mention** are also datasets that we gathered from realistic industrial scenario. Mention tasks are language understanding tasks that model have to judge whether given sentence mentioned sales keywords.

- **Violation** are also datasets that we gathered from realistic industrial scenario. Violation tasks are language understanding tasks that model will have to tell if speaker violates the sales guidelines.

- **Acception** are also datasets that we gathered from realistic industrial scenario. Acception tasks are language understanding tasks that let model tell if the speaker follows systems instruction and tell sales keywords to customer.

- **Execution** are also datasets that we gathered from realistic industrial scenario. Execution tasks are language understanding tasks that model will have to find out whether a salesman follow the predefined sales guidance when talking to customer.

A.1.3 Avoid Test Set Contamination

Although we split datasets into training and testing, there is non-negligible overlap between some of the training datasets and the test set. To avoid test set contamination, we follow the filter method given in (Brown et al., 2020). Specifically, we directly remove all examples in the training phase that have a 30-gram overlap with any example in the test phase.

A.2 Metric

Metrics used for diverse NLP tasks in this paper are presented in the following.

AUC is the abbreviation of Area Under ROC Curve. Typically, the value of AUC is between 0.5 and 1.0.

ROUGE is the abbreviation of Recall-Oriented Understudy for Gisting Evaluation, which is an evaluation method oriented to the recall rate of n-grams. We use ROUGE-1 in the paper.

Micro-F1 is used to evaluate multi-label classification tasks. It is the harmonic average of the averaged precision and recall of all labels.

F1 measures the overlap between the prediction and the ground truth, which is typically used in span prediction tasks.

Pos-F1 is customized for NER tasks with a text-to-text form as shown in Table 16. It is the averaged string F1 score for positive samples, of which the true label is not "blank".

A.3 Build Task-specific Soft Prompts for Unseen Tasks

Firstly, we tune a classifier on the mixture of training data to tell the belongings of given texts, and for new samples in the test task, the classifier can predict the similarities of this sample to training tasks. Formally, for pretrained task i, we regard its task-specific prompt embedding as $\mathcal{E}_i$, the classifier output of training task i's probability as $prob_i$. In our experiments, we have tried three methods to build the test task prompt embedding $\mathcal{E}_{new}$, they are *weighted, top1 and random.*

1) *weighted*. For the *weighted*, we set $\mathcal{E}_{new}$ as a weighted average of pretrained task prompt embedding according to the probability, as

$$\mathcal{E}_{new} = \sum_{i=1}^{N} prob_i \times \mathcal{E}_i \tag{3}$$

We can do the weighted average on sample level, as well as task level.

2) *top1*. For the *top1*, we assign the most similar task prompt embedding to the new task, as

$$\begin{aligned} \mathcal{E}_{new} &= \mathcal{E}_k \\ \text{where } \ k &= \arg\max_i(prob_i),\ i \in N \end{aligned} \tag{4}$$

3) *random*. For the *random*, we initialize the task prompt embedding $\mathcal{E}_{new}$ randomly.

A.4 Auxiliary MLM loss

At the stage of unsupervised pretraining, we apply the span corruption objective, a variant of Masked Language Modeling (MLM), following T5 (Raffel et al., 2020). Meanwhile, we also add MLM as an auxiliary loss to overcome catastrophic forgetting in the multitask pretraining phase.

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{sup} + \mathcal{L}_{\text{MLM}} \tag{5}$$

The multitask pretraining loss is given in Equation 5, where $\mathcal{L}$ is the overall training loss, $\mathcal{L}_{sup}$ is the multitask supervised loss, $\mathcal{L}_{\text{MLM}}$ is the MLM loss and λ is the loss weight. According to Table 18, ZeroPrompt obtains 1.3 point gains by adding the MLM loss, proving our suppose to avoid catastrophic forgetting.

A.5 Data Retrieval and Self-training

To fully exploit unsupervised data, we take a self-training framework similar to (Lee et al., 2013; Du et al., 2021). Given a supervised training set D_{train} and an unlabeled dataset D_{un}, we will retrieve task-similar data from unsupervised corpus according to sentence embedding similarity, and the self-training process may repeat several times. For sentence embedding in retrieval, a pretrained BERT is finetuned on both unsupervised and supervised corpus using SimCSE (Gao et al., 2021b).

The process of self-training is presented in Algorithm 2, where $\mathcal{M}$ is the pretrained model, T is the self-training epoch. For a specific task i, D^i_{train} is the training set and D^i_{un} is the unlabeled dataset. We note D_{train} as the union of all training datasets and D_{un} as the union of all unlabeled datasets.

We select new classification and production datasets to study the impact of data retrieval and self-training, considering similar data available in the unsupervised pretraining corpus. Results are summarized in Table 21. Self-training improves the validation set performance of 0.96 and 0.10 for NEWS and production tasks respectively, and improves the test zero-shot performance of 3.90 and

Algorithm 2 Self-training

Input: $\mathcal{M}$, D_{un}, D_{train}, T
Output: $\mathcal{M}^*$

Init $D^*_{train} \leftarrow D_{train}$
for each $t \in [0, T]$ **do**
  $\mathcal{M}^* \leftarrow$ train M on ${D^*}^i_{train}$
  for each task i **do**
    inference with $\mathcal{M}^*$ on D^i_{un}
    ${D^*}^i_{un} \leftarrow$ select samples in D^i_{un} which are confident with $\mathcal{M}^*$ and make pseudo label,
    $D^*_{train} \leftarrow D^*_{train} \cup {D^*}^i_{un}$,
  end for
end for
return $\mathcal{M}^*$;

1.23. Self-training shows larger improvement on unseen tasks than training tasks. We explain that pseudo labeled data may increase the diversity of training data, resulting better zero-shot generalization abilities.

A.6 Hard Prompt Examples

In this section, we provide details of hard prompts used in this paper. For tasks within each task cluster, we use similar handcrafted prompts as shown in Table 9 $\sim$ 17 . We use both *prefix prompts* and *cloze prompts*. For text classification clusters such as SENTI, NEWS, [X] denotes the sample text. For sentence pair task clusters such as NLI, STS, [X1] denotes the first sample sentence and [X2] is the second sample sentence. For cluster MRC, [X1] denotes the coupus and [X2] denotes the question. For cluster SUM, [X] denotes the coupus, and a similar prompt form is applied for KEYS. For NER, [X1] is the sample text and [X2] denotes the target entity type. For WSC, [X1] is the sample text and [X2] is the pronoun. For all prompts mentioned above, '_' denotes the target position to fill in the answer.

A.7 Detailed Experimental Results

Detailed ablation results of each testing task are presented in Table 18$\sim$21.

Task Type	Task	Train	Test	Metric
Sentiment Analysis (**SENTI**)	yf_amazon	✓		Micro-F1
	JD_full	✓		Micro-F1
	JD_binary	✓		Micro-F1
	waimai_10k	✓		Micro-F1
	online_shopping_10cats		✓	AUC
	ChnSentiCorp_htl_all		✓	AUC
	nlpcc2014_task2		✓	AUC
	weibo_senti_100k		✓	AUC
	yf_dianping		✓	Micro-F1
	car_sentiment		✓	Micro-F1
	dmsc		✓	Micro-F1
	simplifyweibo_4		✓	Micro-F1
	NLPCC2014_Weibo_Emotion_classification		✓	Micro-F1
	nCoV_100k		✓	Micro-F1
	Internet_News		✓	Micro-F1
	BDCI2019		✓	Micro-F1
	SMP2019_ECISA		✓	Micro-F1
News Classification(**NEWS**)	NLPCC2014_LSHT_sample	✓		Micro-F1
	Chinanews	✓		Micro-F1
	CNSS	✓		Micro-F1
	CNSE	✓		Micro-F1
	THUCNews		✓	Micro-F1
	CCFBDCI2020		✓	Micro-F1
	tnews_public		✓	Micro-F1
	Ifeng		✓	Micro-F1
	nlpcc2017_news_headline_categorization		✓	Micro-F1
Intent Classification (**INTENT**)	nlpcc2018_slu	✓		Micro-F1
	catslu_traindev		✓	Micro-F1
	e2e_dials		✓	Micro-F1
	intent_classification		✓	Micro-F1
Natural language inference (**NLI**)	cmnli_public	✓		Micro-F1
	ocnli_public		✓	Micro-F1
Sentence Similarity (**STS**)	LCQMC	✓		AUC
	bq_corpus	✓		AUC
	sohu_sts_A_sl	✓		AUC
	afqmc_public		✓	AUC
	phoenix_pair		✓	AUC
	sohu-sts-A-ll		✓	AUC
	sohu-sts-A-ss		✓	AUC
	sohu-sts-B-ll		✓	AUC
	sohu-sts-B-sl		✓	AUC
	sohu-sts-B-ss		✓	AUC
	CBLUE-CHIP-STS		✓	AUC
	CBLUE-KUAKE-QTR		✓	Micro-F1
	CBLUE-KUAKE-QQR		✓	Micro-F1
Paraphrase (**PARA**)	PAWS-X		✓	AUC
Question Answer Matching (**QAM**)	nlpcc2016-dbqa		✓	AUC
Machine Reading Comprehension Question Answering (**MRC**)	c3_public	✓		F1
	DuReader_robust	✓		F1
	DuReader_checklist	✓		F1
	DuReader_yesno	✓		F1
	dureader	✓		F1
	cmrc2018_public		✓	F1
	DRCD		✓	F1
	CCF2020-BDCI-QA		✓	F1
	CAIL2019-QA		✓	F1
	CAIL2020-QA		✓	F1
Name Entity Recognition (**NER**)	BosonNLP_NER_6C	✓		Pos-F1
	cluener_public	✓		Pos-F1
	RENMIN_NER	✓		Pos-F1
	msra_ner		✓	Pos-F1
	weibo_ner		✓	Pos-F1
	nlpcc2020-AutoIE		✓	Pos-F1
	CCF2020-BDCI-NER		✓	Pos-F1
	CMeEE		✓	Pos-F1
	SanWen-ner		✓	Pos-F1
Summarization (**SUMM**)	LCSTS	✓		ROUGE
	NLPCC2017	✓		ROUGE
	SHENCE	✓		ROUGE
	NLPCC2015		✓	ROUGE
	CAIL2020		✓	ROUGE
	WANFANG		✓	ROUGE
	CSL_SUMM		✓	ROUGE
	EDU_SUMM		✓	ROUGE
	WEIBO		✓	ROUGE
Keywords (**KEYS**)	COTE-BD		✓	F1
	COTE-MFW		✓	F1
	COTE-DP		✓	F1
Winograd Schema Challenge (**WSC**)	cluewsc2020_public		✓	AUC
App Classification (**APP**)	iflytek_public		✓	Micro-F1
Production Datasets	800 datasets for training	✓		AUC
	230 datasets for testing		✓	AUC

Table 8: Summary of collected datasets

Task Type	Prompts	label
Objection	Prompt: 这句话：[X]。上文是否体现了客户对公司不信任？回答： X: 你们是什么公司啊？我从来没听说过你们。 Prompt: This sentence: [X]. Does the customer show objection about the company? Answer: X: What kind of company are yours? I have never heard of it.	是(Yes)/不是(No)
Profile	Prompt: 这句话：[X]。客户是在询问用药后的效果吗？回答： X: 吃了以后的主要作用是什么？。 Prompt: This sentence: [X]. Is the customer asking about the influences of taking the medicine? Answer: X: What is the main effect after taking this?	是(Yes)/不是(No)
Acception	Prompt: 关于电子保单查看，“[X1]”上文销售采纳了与系统推荐“[X2]”相似的描述吗？回答： X1: 让我看一下啊这个您电子版保单这块咱们有接收到吗？ X2: 您的这个电子保单合同有没有收到呢？ Prompt: About electronic insurance policy, Does the salesman say "[X1]" accept the system given expression "[X2]"? Answer: X1: Let me see. Did you received our electronic version of insurance policy? X2: Have you received this electronic policy contract?	采纳(Accept)/ 没有(No)
Violation	Prompt: 这句话：[X]。上文是否体现了坐席私自承诺客户可以随时退款？回答： X: 如果说觉得感觉不太满意的话，你可以直接申请退款。一个月之内，申请退款。 Prompt: This sentence: [X]. Does the customer service privately promise that the customer can refund at any time? Answer: X: If you feel unsatisfied, you can directly apply for a refund. Within one month, apply for a refund.	是(Yes)/不是(No)
Mention	Prompt: 关于保单理赔，“[X1]”是销售提及的内容与文本“[X2]”相似吗？回答： X1: 55种轻症疾病和保险公司达成理赔协议之后7到100个工作日，一次性就把这个钱赔给你了。 X2: 二级及以上公立医院医生的诊断报告啊就可以申请理赔。保险公司呢是直接一次性给到我们100万块钱去看病了。 Prompt: About insurance claim, Does the salesman say "[X1]" mentioned a similar description as "[X2]"? Answer: X1: For 55 mild disease, it will cost 7 to 100 working days after reaching a claim settlement agreement with the insurance company, after that, the money will be paid to you. X2: You can apply for a claim with the diagnosis report of a doctor in a public hospital of level 2 or above. The insurance company will gave you 1 million yuan directly for the disease.	相似(similar)/ 不同(different)
Execution	Prompt: 这句话：[X]。上文坐席是否告知客户存在优惠价格？回答： X: 咱们现在也是有优惠活动的，为何不趁着优惠活动把身体调整一下呢？ Prompt: This sentence: [X]. Does the salesman told customer there are discount price? Answer: X: We have a discount price right now, why not take a change with this discounts?	是(Yes)/不是(No)

Table 9: Illustrations of examples in our production datasets.

Handcrafted Prompt: “[X]”这句汽车评论的态度是什么？_。 Prompt: "[X]", What is the attitude of this car review ?_ X: 动力还可以因为搭载cvt变速箱起步发动机转速比较好。 X: Power can also be equipped with a CVT gearbox to start the engine speed is better. **Augmentation** Prompt: “[X]”如果这个评论的作者是客观的,那么请问,这个评论的内容是什么回答：？_。 Prompt: "[X]", If the author of this comment is objective, what is the content of this comment reply: _ **Verbalizer** 积极(Positive)/消极(Negative)
Target 积极(Positive)

Table 10: Illustrations of prompts in Sentiment Analysis.

Handcrafted
Prompt: 以下这篇新闻是关于什么主题的？_。新闻：[X]
Prompt: What is the topic of the following news?_. News text: [X]
X: 1800万像素单反 佳能60D套机降至9700元 作者：陈 【北京行情】 佳能60D(资料 报价 图片 论坛)是一款拥有1800万像素成像能力，搭载3英寸可翻转LCD显示屏，定位于中端市场的数码单反相机。... 作为佳能畅销单反50D的继承者，佳能EOS 60D对于想拥有一台中端单反相机的用户无疑是一个不错的选择。
X: The Canon 60D is an 18-megapixel digital SLR camera with a 3-inch flip LCD display that is targeted at the mid-market. ... The successor to Canon's best-selling DSLR 50D, the Canon EOS 60D is a good choice for anyone who wants a mid-range DSLR camera.
Augmentation
Prompt: ‘新闻文本’是谁写的?回答：_。“[X]”
Prompt: Who wrote the 'news text'? Answer: _. "[X]"
Verbalizer
科技(Technology)/体育(Sport)/财经(Finance)/娱乐(Entertainment)/..

Target
科技(Technology)

Table 11: Illustrations of prompts in News Classification.

Handcrafted
Prompt: 文章：[X1] 问题：[X2] 回答：_。
Prompt: Corpus: [X1] Question: [X2] Answer:_.
X1: 微信一天最多能转多少钱:这个没有限制吧，到账时间长。纠正下其他网友的回答，微信转账是有限额的。用微信零钱转账最高可以1W元，用银行卡支付就要以银行的额度为准了，最高可以转账5W元。请采纳哦。
X2: 微信一天最多能转多少钱?
X1: Micro letter a day how much money can transfer: there is no limit to it, long to the account. To correct other netizens' answers, wechat transfers are limited. The maximum amount can be 1W yuan with wechat change, and the maximum amount can be 5W yuan for bank card payment. Please adopt it.
X2: How much money can wechat transfer at most a day?
Augmentation
Prompt: 他们是怎么猜出来的?文章：[X1] 问题：[X2] 回答：_。
Prompt: How did they figure that out? Corpus: [X1] Question: [X2] answer: _

Target
微信转账是有限额的。用微信零钱转账最高可以1W元，用银行卡支付就要以银行的额度为准了，最高可以转账5W元
Wechat transfers are limited. The maximum amount can be 1W yuan with wechat change, and the maximum amount can be 5W yuan for bank card payment.

Table 12: Illustrations of prompts in Machine Reading Comprehension Question Answering.

Handcrafted
Prompt: 在通用领域中，第一句话：“[X1]”第二句话：“[X2]”的逻辑关系是什么？回答：_。
Prompt: In the general context, What is the logical relationship between the first sentence "[X1]" and the second sentence "[X2]". Answer: _.
X1: 等他回来,我们就出去吃啊。
X1: When he gets back, we'll eat out.
X2: 我们在等他。
X2: We are waiting for him.
Augmentation
Prompt: 这两句话是如何组合在一起的?回答：_。第一句话：“[X1]”，第二句话：“[X2]”
Prompt: How do these two sentences go together? Answer: _. the first sentence: "[X1]", the second sentence: "[X2]".
Verbalizer
相反(contradiction)/中性(neutral)/一致(entailment)

Target
一致(entailment)

Table 13: Illustrations of prompts in Natural Language Inference.

Handcrafted
Prompt: 在金融领域中，第一句话：“[X1]”第二句话：“[X2]”这两句话含义 _。
Prompt: In finance context, the first sentence: "[X1]" the second sentence: "[X2]", the meaning of these two sentences is _.
X1: 花呗支持高铁票支付吗?
X1: Does Huabei support high-speed rail ticket payment?
X2: 为什么不支持花呗付款?
X2: Why not support the payment of Huabei?
Augmentation
Prompt: 它们之间的关系是怎样的?回答：_。第一句话：“[X1]”，第二句话：“[X2]”
Prompt: What is the relationship between them? Answer: _. the first sentence: "[X1]", the second sentence: "[X2]".
Verbalizer
相似(similar)/不同(different)
Target
不同(different)

Table 14: Illustrations of prompts in Sentence Similarity.

Handcrafted
Prompt: 对于句子：[X1] 代词：[X2] 指代的是：[X3] 吗？回答：_。
Prompt: In the sentence: [X1], does the pronoun [X2] refer to [X3]? Answer: _.
X1: 满银的老祖上曾经当过“拔贡”。先人手里在这一带有过些名望。到他祖父这代就把一点家业败光了。
X2: 他
X3: 满银
X1: The old ancestor of Manyin used to be "baogong". There was some renown in the hands of our ancestors. By his grandfather's generation the family business had been wiped out.
X2: he
X3: Manyin
Augmentation
Prompt: 第二句话中,有两个“它”：[X1] 其中：[X2]指的_[X3]。
Prompt: In the second sentence, there are two "it" s: [X1] among this sentence: [X2] refer to [X3]? _
Verbalizer
是(yes)/不是(no)
Target
是(yes)

Table 15: Illustrations of prompts in Winograd Schema Chanllenge.

Handcrafted
Prompt: 报纸文本：[X1]中有哪些属于[X2]？回答
Prompt: Text from newspaper : Which words of [X1] belong to [X2]? Answer: _.
X1: 相比之下，青岛海牛队和广州松日队的雨中之战虽然也是0：0，但乏善可陈。
X2: 机构名
X1: In contrast, although the raining war between Qingdao manatee team and Guangzhou songri team is also 0:0, but it is too lackluster.
X2: organization
Augmentation
Prompt: 回答：_。文本[X1] 报纸文本中的[X2]类别的实体是由哪些部分构成的?
Prompt: Answer: _. Text from newspaper: [X1]. Which parts make up the entities of the [X2] category in newspaper text?
Target
青岛海牛队，广州松日队
Qingdao manatee team, Guangzhou songri team

Table 16: Illustrations of prompts in Name Entity Recognition. Each example is extended to N instances, where N is the number of possible entity type. For each entity type, we ask the model to predict corresponding entities presented in the given text. The ground truth is "blank" if there is no entity of that type in the sentence.

Handcrafted
Prompt: [X]，这个教育相关的文本的摘要为：_。
Prompt: [X], A summary of this education-related text: _.
X: 中新网2月25日电 据外媒报道，意大利一名小女孩嘉比是一位动物爱好者，她经常拿自己的零食和家里的剩菜喂乌鸦，因此而收到了乌鸦送的“礼物”。据报道，嘉比经常用花生、狗粮和一些剩菜喂乌鸦，她表示，自己不是为了获得奖励而做这些，而是因为她喜欢自然。最近，乌鸦经常衔一些亮晶晶的东西给她，里面通常是些钮扣、文具和五金之类的小东西，有几次她还收到耳环，乌鸦甚至帮她妈妈把遗失的相机盖找了回去。禽鸟专家表示，乌鸦确实有和人类交朋友的能力，所以乌鸦报恩不是小女孩的想象。
X: China News on February 25: Gabi, an Italian girl who loves animals, has received a gift from a crow for feeding her snacks and family leftovers, foreign media reported. Gaby reportedly regularly feeds the crows peanuts, dog food and some leftovers, and she said she does not ask a reward but because she loves nature. Lately, they've been bringing her shiny things, usually buttons, stationery and hardware. In a few cases, she's received earrings. They even helped her mother find the cover of a camera she'd lost. According to bird experts, crows do have the ability to make friends with humans, so it's not a little girl's imagination for them to return the favor.
Augmentation
Prompt: [X] 这个领域的领域词典中收录的单词,应该是_。
Prompt: [X] The words in the domain dictionary of this field should be _.

Target
意大利女童用零食喂乌鸦，乌鸦送“礼物”报恩"
Talian girl feeds snacks to crows who return kindness with 'gifts'

Table 17: Illustrations of prompts in Summarization.

Model	- $\mathcal{E},\mathcal{V}$	- $\mathcal{V}$	- $\mathcal{E}$	ZeroPrompt	+ MLM
Total Scores*	$42.27_{(0.34)}$	$42.88_{(0.55)}$	$45.06_{(0.69)}$	$46.16_{(0.54)}$	$47.43_{(0.76)}$
online_shopping_10cats	$96.11_{(0.31)}$	$96.06_{(0.27)}$	$95.55_{(0.31)}$	$95.72_{(0.27)}$	$95.90_{(0.24)}$
ChnSentiCorp_htl_all	$93.80_{(0.51)}$	$93.75_{(0.57)}$	$93.44_{(0.47)}$	$93.45_{(0.38)}$	$93.98_{(0.55)}$
nlpcc2014_task2	$79.05_{(0.81)}$	$80.42_{(0.49)}$	$80.28_{(0.64)}$	$80.12_{(0.24)}$	$80.49_{(0.41)}$
yf_dianping	$37.27_{(2.66)}$	$37.27_{(3.85)}$	$45.11_{(5.41)}$	$44.87_{(4.48)}$	$43.89_{(2.51)}$
car_sentiment	$23.98_{(0.57)}$	$30.49_{(5.57)}$	$24.38_{(1.64)}$	$25.80_{(3.41)}$	$25.63_{(1.70)}$
dmsc	$34.25_{(2.13)}$	$36.94_{(2.65)}$	$37.16_{(3.73)}$	$37.88_{(2.31)}$	$36.97_{(3.08)}$
weibo_senti_100k	$86.48_{(0.58)}$	$86.39_{(1.99)}$	$84.23_{(1.00)}$	$85.89_{(1.22)}$	$86.48_{(1.55)}$
simplifyweibo_4	$18.70_{(2.20)}$	$20.38_{(2.23)}$	$44.58_{(1.20)}$	$38.87_{(2.06)}$	$42.66_{(4.60)}$
NLPCC2014_Weibo_Emotion_classification	$37.57_{(1.39)}$	$38.90_{(1.20)}$	$40.56_{(0.93)}$	$41.21_{(1.08)}$	$41.28_{(1.69)}$
nCoV_100k	$34.11_{(0.53)}$	$33.62_{(1.59)}$	$33.20_{(2.00)}$	$34.82_{(1.35)}$	$34.91_{(0.49)}$
Internet_News	$53.61_{(2.23)}$	$48.99_{(1.95)}$	$52.42_{(10.39)}$	$55.20_{(8.58)}$	$56.92_{(2.78)}$
BDCI2019	$26.91_{(5.09)}$	$22.53_{(3.45)}$	$29.75_{(5.22)}$	$36.53_{(5.45)}$	$32.81_{(3.04)}$
SMP2019_ECISA	$38.18_{(1.25)}$	$36.44_{(1.51)}$	$35.71_{(2.76)}$	$38.44_{(1.87)}$	$38.46_{(0.33)}$
THUCNews	$47.43_{(2.77)}$	$51.45_{(3.98)}$	$66.06_{(2.14)}$	$65.86_{(2.93)}$	$68.66_{(1.29)}$
CCFBDCI2020	$71.92_{(0.98)}$	$69.54_{(3.55)}$	$74.78_{(4.00)}$	$75.93_{(4.21)}$	$80.50_{(1.68)}$
tnews_public	$35.10_{(1.14)}$	$34.23_{(3.66)}$	$46.67_{(1.49)}$	$46.35_{(1.50)}$	$49.90_{(1.36)}$
Ifeng	$60.41_{(1.97)}$	$57.96_{(4.12)}$	$61.32_{(0.94)}$	$62.79_{(1.21)}$	$63.04_{(2.27)}$
nlpcc2017_news_headline_categorization	$33.00_{(1.67)}$	$33.52_{(2.52)}$	$47.56_{(1.72)}$	$47.14_{(1.37)}$	$50.26_{(1.43)}$
catslu_traindev	$90.79_{(0.56)}$	$91.59_{(0.80)}$	$90.45_{(0.43)}$	$91.33_{(0.54)}$	$90.48_{(0.78)}$
e2e_dials	$69.20_{(2.92)}$	$67.27_{(4.14)}$	$82.02_{(2.02)}$	$86.39_{(5.50)}$	$88.44_{(5.28)}$
intent_classification	$20.41_{(1.05)}$	$24.99_{(0.52)}$	$28.47_{(1.47)}$	$34.37_{(4.38)}$	$33.64_{(3.84)}$
ocnli_public	$45.60_{(1.19)}$	$47.60_{(0.16)}$	$47.70_{(1.20)}$	$47.16_{(2.09)}$	$46.16_{(1.87)}$
afqmc_public	$63.40_{(0.79)}$	$64.37_{(0.57)}$	$63.63_{(0.91)}$	$63.52_{(0.88)}$	$64.60_{(0.49)}$
phoenix_pair	$98.90_{(0.22)}$	$99.28_{(0.30)}$	$98.77_{(0.44)}$	$98.99_{(0.17)}$	$98.99_{(0.24)}$
sohu-sts-A-ll	$64.65_{(0.60)}$	$64.04_{(0.97)}$	$64.21_{(0.50)}$	$65.44_{(0.72)}$	$65.92_{(0.78)}$
sohu-sts-A-ss	$70.91_{(0.37)}$	$71.83_{(1.56)}$	$69.88_{(1.34)}$	$70.70_{(0.74)}$	$70.80_{(0.46)}$
sohu-sts-B-ll	$60.32_{(1.69)}$	$60.03_{(1.15)}$	$60.69_{(1.24)}$	$62.23_{(1.70)}$	$61.47_{(0.79)}$
sohu-sts-B-sl	$65.56_{(1.69)}$	$64.51_{(1.08)}$	$68.08_{(3.01)}$	$68.76_{(3.09)}$	$70.34_{(0.84)}$
sohu-sts-B-ss	$77.61_{(1.82)}$	$80.05_{(0.86)}$	$79.64_{(0.80)}$	$80.03_{(0.97)}$	$79.85_{(1.03)}$
CBLUE-CHIP-STS	$75.80_{(1.21)}$	$76.90_{(0.62)}$	$75.91_{(1.12)}$	$75.69_{(0.38)}$	$77.90_{(0.59)}$
CBLUE-KUAKE-QTR	$26.75_{(0.57)}$	$27.00_{(0.56)}$	$25.97_{(1.28)}$	$26.11_{(0.77)}$	$25.35_{(1.60)}$
CBLUE-KUAKE-QQR	$43.57_{(2.03)}$	$41.79_{(3.05)}$	$38.47_{(7.19)}$	$41.74_{(5.35)}$	$35.35_{(8.27)}$
PAWS-X	$53.52_{(0.64)}$	$55.14_{(0.71)}$	$54.19_{(0.59)}$	$54.41_{(0.99)}$	$54.90_{(0.37)}$
nlpcc2016-dbqa	$63.89_{(2.07)}$	$60.90_{(0.44)}$	$64.24_{(2.68)}$	$62.77_{(0.80)}$	$62.61_{(3.64)}$
cmrc2018_public	$32.78_{(2.01)}$	$33.24_{(2.70)}$	$34.86_{(2.32)}$	$32.07_{(1.51)}$	$35.50_{(0.73)}$
DRCD	$44.31_{(3.45)}$	$43.08_{(2.69)}$	$44.81_{(2.27)}$	$43.11_{(1.91)}$	$47.89_{(2.20)}$
CCF2020-BDCI-QA	$13.05_{(1.13)}$	$13.86_{(1.73)}$	$15.27_{(0.91)}$	$15.15_{(0.49)}$	$16.22_{(0.56)}$
CAIL2019-QA	$22.25_{(1.16)}$	$21.31_{(1.11)}$	$23.20_{(0.67)}$	$20.61_{(1.48)}$	$22.84_{(1.61)}$
CAIL2020-QA	$27.90_{(1.48)}$	$24.84_{(3.29)}$	$26.45_{(1.50)}$	$23.64_{(0.81)}$	$26.87_{(2.14)}$
msra_ner	$57.18_{(4.84)}$	$55.38_{(6.00)}$	$57.88_{(5.04)}$	$60.07_{(3.97)}$	$58.17_{(4.40)}$
weibo_ner	$22.71_{(1.95)}$	$23.24_{(0.95)}$	$23.16_{(1.42)}$	$23.28_{(1.62)}$	$23.42_{(0.52)}$
nlpcc2020-AutoIE	$33.65_{(6.85)}$	$30.82_{(3.52)}$	$33.95_{(3.15)}$	$37.17_{(4.88)}$	$35.29_{(6.25)}$
CCF2020-BDCI-NER	$46.83_{(2.91)}$	$45.45_{(3.76)}$	$48.46_{(2.37)}$	$47.35_{(3.30)}$	$47.34_{(2.30)}$
CMeEE	$24.87_{(3.15)}$	$21.60_{(2.08)}$	$25.59_{(3.58)}$	$23.93_{(3.09)}$	$24.84_{(0.94)}$
SanWen-ner	$18.31_{(1.96)}$	$16.72_{(1.79)}$	$19.13_{(2.85)}$	$17.82_{(1.96)}$	$18.42_{(1.63)}$
NLPCC2015	$2.46_{(0.33)}$	$2.47_{(0.47)}$	$2.37_{(0.27)}$	$2.45_{(0.46)}$	$2.78_{(0.33)}$
CAIL2020	$0.86_{(0.16)}$	$0.60_{(0.16)}$	$0.82_{(0.32)}$	$0.77_{(0.41)}$	$0.81_{(0.05)}$
WANFANG	$5.25_{(0.24)}$	$5.23_{(0.81)}$	$5.44_{(0.36)}$	$5.46_{(0.42)}$	$7.00_{(0.22)}$
CSL_SUMM	$1.48_{(0.22)}$	$1.82_{(0.26)}$	$1.74_{(0.47)}$	$2.05_{(0.30)}$	$3.35_{(0.55)}$
EDU_SUMM	$15.50_{(4.52)}$	$14.74_{(1.89)}$	$18.72_{(0.95)}$	$15.04_{(2.67)}$	$14.80_{(3.15)}$
WEIBO	$4.95_{(0.94)}$	$5.41_{(0.31)}$	$4.95_{(0.67)}$	$4.66_{(0.65)}$	$5.45_{(0.45)}$
COTE-BD	$6.81_{(1.61)}$	$23.61_{(7.55)}$	$20.79_{(3.38)}$	$40.58_{(6.56)}$	$48.29_{(9.36)}$
COTE-MFW	$14.38_{(2.46)}$	$32.34_{(9.76)}$	$25.14_{(4.61)}$	$43.81_{(6.53)}$	$50.34_{(9.01)}$
COTE-DP	$7.94_{(3.72)}$	$18.46_{(9.97)}$	$21.07_{(4.50)}$	$23.89_{(10.29)}$	$42.50_{(6.43)}$
cluewsc2020_public	$45.66_{(2.39)}$	$42.76_{(1.40)}$	$40.26_{(1.97)}$	$42.06_{(1.35)}$	$47.98_{(4.18)}$
iflytek_public	$18.99_{(2.70)}$	$18.22_{(2.51)}$	$23.95_{(3.17)}$	$23.45_{(3.49)}$	$26.14_{(1.02)}$

Table 18: Detailed ablation results on prompt design and MLM loss

	none	weighted avg all samples	weighted avg per sample	top1 per sample	random init
ALL	$44.83_{(0.55)}$	$45.76_{(0.42)}$	$46.01_{(0.52)}$	$46.06_{(0.55)}$	$46.16_{(0.54)}$
online_shopping_10cats	$95.49_{(0.30)}$	$95.73_{(0.27)}$	$95.73_{(0.27)}$	$95.73_{(0.27)}$	$95.72_{(0.27)}$
ChnSentiCorp_htl_all	$92.92_{(0.51)}$	$93.51_{(0.37)}$	$93.42_{(0.37)}$	$93.43_{(0.35)}$	$93.45_{(0.38)}$
nlpcc2014_task2	$79.90_{(0.29)}$	$80.14_{(0.24)}$	$80.14_{(0.23)}$	$80.13_{(0.24)}$	$80.12_{(0.24)}$
yf_dianping	$44.80_{(4.49)}$	$44.63_{(4.68)}$	$44.66_{(4.65)}$	$44.63_{(4.66)}$	$44.87_{(4.48)}$
car_sentiment	$24.44_{(1.81)}$	$25.74_{(3.38)}$	$25.73_{(3.37)}$	$25.79_{(3.37)}$	$25.80_{(3.41)}$
dmsc	$38.21_{(2.38)}$	$37.77_{(2.48)}$	$37.81_{(2.30)}$	$37.90_{(2.27)}$	$37.88_{(2.31)}$
weibo_senti_100k	$85.21_{(1.31)}$	$85.45_{(0.94)}$	$85.95_{(1.22)}$	$85.91_{(1.23)}$	$85.89_{(1.22)}$
simplifyweibo_4	$39.54_{(3.07)}$	$38.01_{(1.78)}$	$38.67_{(1.76)}$	$38.78_{(1.79)}$	$38.87_{(2.06)}$
NLPCC2014_Weibo_Emotion_classification	$40.41_{(1.06)}$	$41.23_{(1.18)}$	$41.19_{(0.87)}$	$41.22_{(0.94)}$	$41.21_{(1.08)}$
nCoV_100k	$34.46_{(1.51)}$	$34.86_{(1.32)}$	$34.80_{(1.34)}$	$34.82_{(1.38)}$	$34.82_{(1.35)}$
Internet_News	$55.32_{(8.07)}$	$55.12_{(8.58)}$	$55.10_{(8.55)}$	$55.19_{(8.58)}$	$55.20_{(8.58)}$
BDCI2019	$35.69_{(5.31)}$	$36.29_{(5.45)}$	$36.46_{(5.43)}$	$36.52_{(5.42)}$	$36.53_{(5.45)}$
SMP2019_ECISA	$37.63_{(2.15)}$	$38.49_{(1.90)}$	$38.51_{(1.88)}$	$38.51_{(1.87)}$	$38.44_{(1.87)}$
THUCNews	$65.58_{(3.27)}$	$65.90_{(2.91)}$	$65.89_{(2.91)}$	$65.87_{(2.91)}$	$65.86_{(2.93)}$
CCFBDCI2020	$75.61_{(4.08)}$	$75.98_{(3.87)}$	$75.86_{(4.13)}$	$75.83_{(4.20)}$	$75.93_{(4.21)}$
tnews_public	$46.04_{(1.26)}$	$46.42_{(1.38)}$	$46.36_{(1.42)}$	$46.32_{(1.42)}$	$46.35_{(1.50)}$
Ifeng	$63.66_{(1.44)}$	$62.78_{(1.20)}$	$62.77_{(1.21)}$	$62.77_{(1.18)}$	$62.79_{(1.21)}$
nlpcc2017_news_headline_categorization	$46.95_{(1.36)}$	$47.15_{(1.27)}$	$47.16_{(1.31)}$	$47.14_{(1.29)}$	$47.14_{(1.37)}$
catslu_traindev	$90.55_{(0.74)}$	$91.52_{(0.39)}$	$91.57_{(0.42)}$	$91.52_{(0.39)}$	$91.33_{(0.54)}$
e2e_dials	$88.24_{(5.05)}$	$86.38_{(5.55)}$	$86.36_{(5.50)}$	$86.44_{(5.53)}$	$86.39_{(5.50)}$
intent_classification	$32.04_{(3.89)}$	$34.37_{(4.37)}$	$34.34_{(4.39)}$	$34.37_{(4.37)}$	$34.37_{(4.38)}$
ocnli_public	$46.98_{(1.96)}$	$47.34_{(1.99)}$	$47.21_{(2.06)}$	$47.17_{(2.01)}$	$47.16_{(2.09)}$
afqmc_public	$62.96_{(0.92)}$	$63.51_{(0.87)}$	$63.50_{(0.86)}$	$63.50_{(0.86)}$	$63.52_{(0.88)}$
phoenix_pair	$97.92_{(0.98)}$	$98.99_{(0.20)}$	$98.98_{(0.20)}$	$98.99_{(0.20)}$	$98.99_{(0.17)}$
sohu-sts-A-ll	$64.97_{(0.57)}$	$65.47_{(0.72)}$	$65.47_{(0.73)}$	$65.46_{(0.72)}$	$65.44_{(0.72)}$
sohu-sts-A-ss	$70.19_{(0.89)}$	$70.80_{(0.67)}$	$70.73_{(0.70)}$	$70.72_{(0.74)}$	$70.70_{(0.74)}$
sohu-sts-B-ll	$61.81_{(1.39)}$	$62.23_{(1.64)}$	$62.22_{(1.61)}$	$62.22_{(1.64)}$	$62.23_{(1.70)}$
sohu-sts-B-sl	$68.48_{(2.57)}$	$68.77_{(3.11)}$	$68.77_{(3.11)}$	$68.76_{(3.11)}$	$68.76_{(3.09)}$
sohu-sts-B-ss	$79.77_{(0.78)}$	$80.00_{(0.99)}$	$79.99_{(0.94)}$	$80.01_{(0.96)}$	$80.03_{(0.97)}$
CBLUE-CHIP-STS	$74.93_{(0.51)}$	$75.66_{(0.36)}$	$75.67_{(0.36)}$	$75.67_{(0.36)}$	$75.69_{(0.38)}$
CBLUE-KUAKE-QTR	$25.73_{(0.85)}$	$26.11_{(0.85)}$	$26.14_{(0.86)}$	$26.12_{(0.84)}$	$26.11_{(0.77)}$
CBLUE-KUAKE-QQR	$41.09_{(6.06)}$	$41.62_{(5.20)}$	$41.70_{(5.22)}$	$41.62_{(5.21)}$	$41.74_{(5.35)}$
PAWS-X	$54.48_{(1.11)}$	$54.39_{(0.96)}$	$54.40_{(0.96)}$	$54.39_{(0.96)}$	$54.41_{(0.99)}$
nlpcc2016-dbqa	$59.45_{(2.65)}$	$62.86_{(0.87)}$	$62.81_{(0.93)}$	$62.84_{(0.87)}$	$62.77_{(0.80)}$
cmrc2018_public	$34.43_{(1.64)}$	$32.00_{(1.54)}$	$31.94_{(1.54)}$	$31.90_{(1.54)}$	$32.07_{(1.51)}$
DRCD	$42.99_{(3.90)}$	$42.48_{(2.52)}$	$42.57_{(2.50)}$	$42.50_{(2.50)}$	$43.11_{(1.91)}$
CCF2020-BDCI-QA	$16.20_{(1.02)}$	$14.96_{(0.53)}$	$14.99_{(0.54)}$	$15.15_{(0.69)}$	$15.15_{(0.49)}$
CAIL2019-QA	$20.88_{(2.19)}$	$20.29_{(1.32)}$	$20.52_{(1.47)}$	$20.58_{(1.54)}$	$20.61_{(1.48)}$
CAIL2020-QA	$22.62_{(2.14)}$	$23.29_{(0.84)}$	$23.43_{(0.61)}$	$23.61_{(0.63)}$	$23.64_{(0.81)}$
msra_ner	$60.67_{(4.12)}$	$60.05_{(4.45)}$	$60.08_{(4.30)}$	$60.00_{(4.13)}$	$60.07_{(3.97)}$
weibo_ner	$23.20_{(1.60)}$	$23.36_{(1.72)}$	$23.47_{(1.80)}$	$23.48_{(1.72)}$	$23.28_{(1.62)}$
nlpcc2020-AutoIE	$38.95_{(6.31)}$	$35.92_{(4.59)}$	$36.88_{(4.98)}$	$36.78_{(4.95)}$	$37.17_{(4.88)}$
CCF2020-BDCI-NER	$47.51_{(4.18)}$	$47.28_{(3.68)}$	$47.35_{(3.40)}$	$47.47_{(3.31)}$	$47.35_{(3.30)}$
CMeEE	$21.25_{(2.78)}$	$24.26_{(3.27)}$	$24.18_{(3.23)}$	$23.80_{(3.11)}$	$23.93_{(3.09)}$
SanWen-ner	$18.26_{(1.91)}$	$17.80_{(2.06)}$	$17.85_{(2.03)}$	$17.90_{(1.93)}$	$17.82_{(1.96)}$
NLPCC2015	$2.05_{(0.33)}$	$2.41_{(0.42)}$	$2.37_{(0.44)}$	$2.55_{(0.44)}$	$2.45_{(0.46)}$
CAIL2020	$0.79_{(0.39)}$	$0.74_{(0.42)}$	$0.77_{(0.42)}$	$0.81_{(0.45)}$	$0.77_{(0.41)}$
WANFANG	$5.64_{(0.52)}$	$5.30_{(0.38)}$	$5.32_{(0.32)}$	$5.39_{(0.47)}$	$5.46_{(0.42)}$
CSL_SUMM	$1.69_{(0.37)}$	$1.89_{(0.25)}$	$1.84_{(0.24)}$	$1.91_{(0.33)}$	$2.05_{(0.30)}$
EDU_SUMM	$16.81_{(1.73)}$	$13.71_{(2.73)}$	$14.80_{(2.94)}$	$15.10_{(2.87)}$	$15.04_{(2.67)}$
WEIBO	$5.40_{(0.88)}$	$4.61_{(0.62)}$	$4.63_{(0.62)}$	$4.68_{(0.65)}$	$4.66_{(0.65)}$
COTE-BD	$14.62_{(4.81)}$	$26.80_{(4.97)}$	$38.13_{(6.50)}$	$39.09_{(7.09)}$	$40.58_{(6.56)}$
COTE-MFW	$16.35_{(5.31)}$	$41.65_{(8.03)}$	$40.64_{(7.40)}$	$41.65_{(7.63)}$	$43.81_{(6.53)}$
COTE-DP	$12.21_{(7.17)}$	$22.62_{(10.85)}$	$22.69_{(10.79)}$	$22.80_{(11.12)}$	$23.89_{(10.29)}$
cluewsc2020_public	$43.11_{(0.63)}$	$42.50_{(1.41)}$	$42.50_{(1.41)}$	$42.50_{(1.41)}$	$42.06_{(1.35)}$
iflytek_public	$23.61_{(3.30)}$	$23.39_{(3.50)}$	$23.39_{(3.51)}$	$23.37_{(3.41)}$	$23.45_{(3.49)}$

Table 19: Detailed ablation results on building new task-specific soft prompts

Methods	LM-BFF	LM-BFF†	BT	PTMs (GPT2)	PTMs (CPM)	BT&GPS (ours)	PTMs(GPT2) &GPS (ours)	PTMs(CPM) &GPS (ours)
Avg	43.45	44.45	45.76	46.03	46.53	45.91	46.64	47.05
online_shopping_10cats	94.62	95.28	95.56	95.69	95.39	95.42	94.87	95.34
ChnSentiCorp_htl_all	92.66	92.98	93.57	93.32	93.33	93.67	92.62	93.53
nlpcc2014_task2	79.37	80.01	80.38	79.87	80.31	80.25	79.46	78.34
yf_dianping	44.01	46.29	45.45	42.83	45.70	46.03	45.20	45.11
car_sentiment	28.52	24.84	28.80	27.30	30.18	25.06	22.97	28.58
dmsc	36.57	36.54	37.07	36.87	37.11	36.10	35.70	36.83
weibo_senti_100k	83.17	83.45	84.15	84.41	84.10	84.19	83.62	85.03
simplifyweibo_4	43.84	44.29	41.24	38.38	44.94	39.56	37.42	42.89
NLPCC2014_Weibo_Emotion_classification	38.68	39.51	40.72	39.32	39.63	41.13	37.31	39.95
nCoV_100k	32.46	32.78	33.18	33.17	33.08	27.92	33.61	31.30
Internet_News	51.75	52.21	52.91	53.28	52.82	52.91	53.28	53.07
BDCI2019	27.36	29.69	29.51	32.89	30.41	30.44	32.89	31.85
SMP2019_ECISA	33.76	35.24	35.56	35.61	36.07	39.22	34.65	36.07
THUCNews	65.42	66.03	66.00	65.53	65.69	65.54	65.53	65.69
CCFBDCI2020	74.70	74.63	74.60	74.34	74.98	74.60	72.50	75.39
tnews_public	46.63	46.85	46.70	45.58	46.52	46.71	45.56	46.45
Ifeng	60.86	60.80	61.26	62.16	61.16	62.31	62.07	62.17
nlpcc2017_news_headline_categorization	46.01	47.06	47.61	47.72	47.72	47.55	47.72	47.20
catslu_traindev	88.59	89.09	90.49	90.55	90.70	89.45	89.42	89.80
e2e_dials	82.77	83.25	82.82	83.11	84.92	85.88	83.32	88.65
intent_classification	27.26	29.10	30.03	33.15	31.52	30.41	33.36	32.25
ocnli_public	45.64	45.71	47.45	47.89	47.60	44.78	47.02	47.56
afqmc_public	61.48	62.82	63.04	63.49	63.82	60.58	63.30	63.19
phoenix_pair	94.64	98.03	98.69	98.58	98.81	99.01	98.35	98.81
sohu-sts-A-ll	63.76	64.05	63.92	63.68	63.79	63.73	63.28	63.80
sohu-sts-A-ss	69.01	69.83	70.68	66.56	70.33	67.68	69.11	69.69
sohu-sts-B-ll	59.91	60.29	60.22	60.66	60.72	59.86	57.08	59.11
sohu-sts-B-sl	64.75	66.92	68.01	67.76	68.11	68.11	67.50	68.97
sohu-sts-B-ss	75.95	78.80	79.39	79.39	79.60	78.47	78.51	79.88
CBLUE-CHIP-STS	73.88	74.77	75.62	75.12	75.85	75.41	76.09	74.52
CBLUE-KUAKE-QTR	25.22	25.76	26.30	26.43	26.31	25.28	26.32	26.31
CBLUE-KUAKE-QQR	37.45	36.30	37.84	37.64	38.40	43.59	42.59	38.40
PAWS-X	53.13	54.21	54.68	53.66	54.26	55.12	53.70	54.47
nlpcc2016-dbqa	61.98	62.71	63.46	65.75	65.82	63.69	63.61	65.68
cmrc2018_public	28.82	32.43	35.94	35.76	35.87	35.56	35.75	35.73
DRCD	29.58	40.07	45.98	47.37	47.97	45.17	47.64	47.14
CCF2020-BDCI-QA	13.10	14.71	15.48	15.61	15.70	15.43	15.18	15.55
CAIL2019-QA	11.28	20.13	23.32	23.17	23.28	23.35	22.37	23.21
CAIL2020-QA	17.54	22.73	25.38	26.35	26.06	23.67	25.99	25.31
msra_ner	48.23	27.32	59.98	59.22	59.34	59.94	59.64	58.36
weibo_ner	14.72	21.83	22.07	23.27	23.13	24.53	19.31	23.40
nlpcc2020-AutoIE	27.12	36.68	36.45	45.83	44.01	35.00	49.05	49.14
CCF2020-BDCI-NER	42.26	44.26	48.14	46.80	48.45	48.29	46.80	48.45
CMeEE	16.52	21.20	25.09	21.40	23.91	25.03	20.05	23.91
SanWen-ner	15.93	16.46	19.22	21.49	19.06	17.73	20.92	20.58
NLPCC2015	2.25	2.30	2.37	2.34	2.34	2.23	2.33	2.42
CAIL2020	0.85	0.84	0.82	0.80	0.84	0.82	0.77	0.83
WANFANG	5.56	5.93	5.65	5.47	5.17	5.79	5.23	4.17
CSL_SUMM	2.42	1.59	1.74	1.74	1.40	0.89	1.14	1.40
EDU_SUMM	20.42	20.27	20.71	20.20	22.19	22.64	21.74	21.48
WEIBO	4.51	4.73	4.67	3.92	4.04	3.96	4.21	5.42
COTE-BD	36.80	38.44	38.24	44.94	44.63	47.76	49.43	49.97
COTE-MFW	35.86	29.15	32.75	38.17	44.98	41.94	55.19	47.24
COTE-DP	31.06	34.75	28.54	27.22	26.51	25.03	34.41	45.49
cluewsc2020_public	37.19	38.22	37.46	39.76	41.71	41.16	59.96	45.09
iflytek_public	25.25	24.82	25.35	25.07	25.55	25.43	25.29	24.74

Table 20: Detailed ablation results on genetic prompt search

Train Task	baseline dev	self-training dev	Test Task	baseline test	self-training test
NEWS AVG(4)	**86.49**	**87.45 (↑0.96)**	**NEWS AVG(5)**	**55.21**	**59.11 (↑3.90)**
NLPCC2014_LSHT	80.10	82.83 (↑2.73)	CCFBDCI2020	73.97	75.48 (↑1.51)
CNSE	92.68	93.36 (↑0.68)	THUCNews	61.55	68.14 (↑6.59)
CNSS	93.77	93.81 (↑0.04)	Ifeng	63.43	64.18 (↑0.75)
Chinanews	79.42	79.79 (↑0.37)	nlpcc2017_news	38.57	43.32 (↑4.75)
			tnews_public	38.54	44.46 (↑5.92)
production AVG(122)	**81.84**	**81.94 (↑0.10)**	**production AVG(130)**	**78.08**	**79.31 (↑1.23)**
cluster1(42)	83.50	84.18 (↑0.68)	cluster5(14)	77.65	79.65 (↑2.00)
cluster2(8)	80.13	79.45 (↓0.68)	cluster6(68)	78.83	79.62 (↑0.79)
cluster3(22)	81.58	81.16 (↓0.42)	cluster7(26)	82.5	84.09 (↑1.59)
cluster4(50)	80.90	80.77 (↓0.13)	cluster8(22)	70.81	72.46 (↑1.65)

Table 21: Detailed results on retrieval and self-training